Modelos machine learning

Javier Saavedra

Introduccion

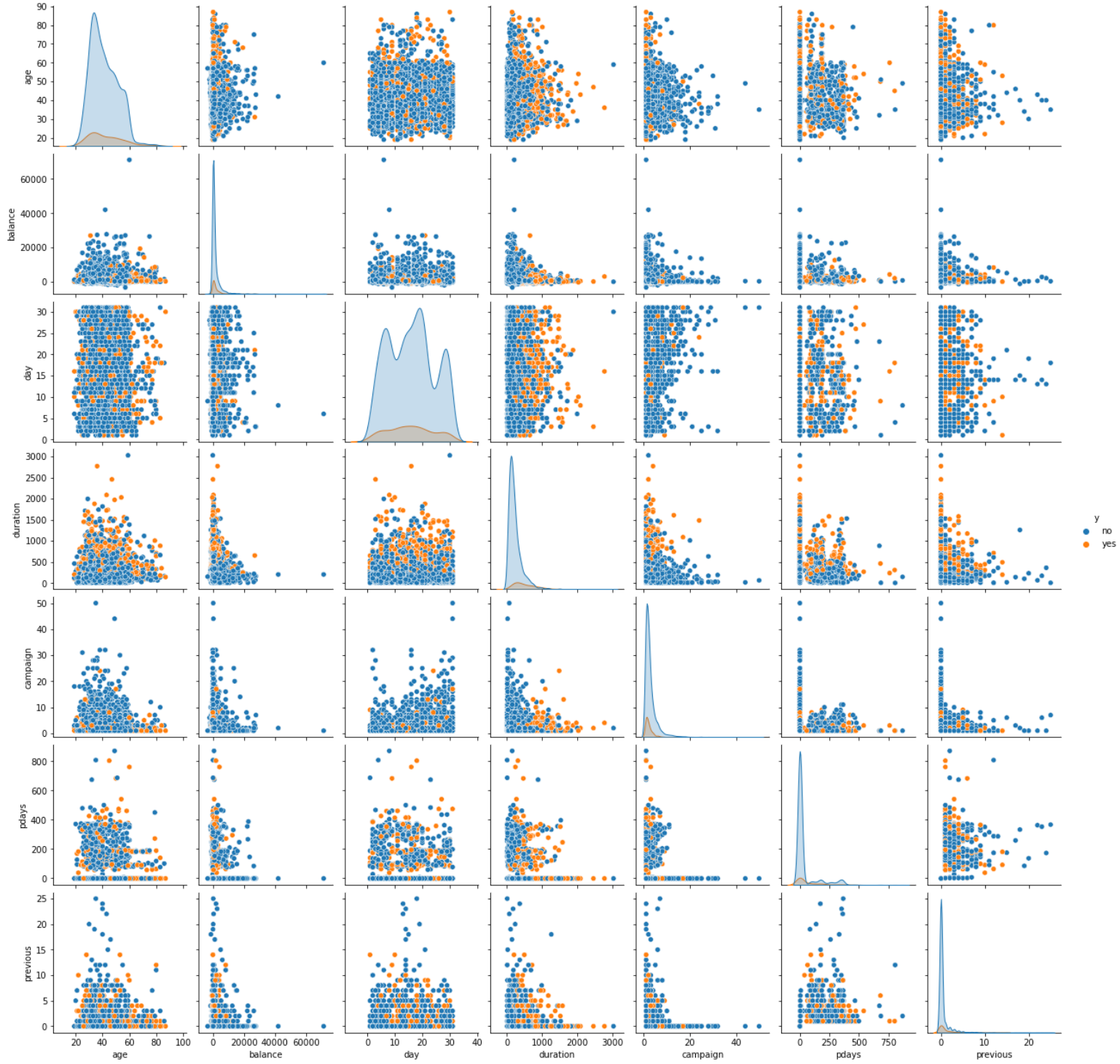
Datos

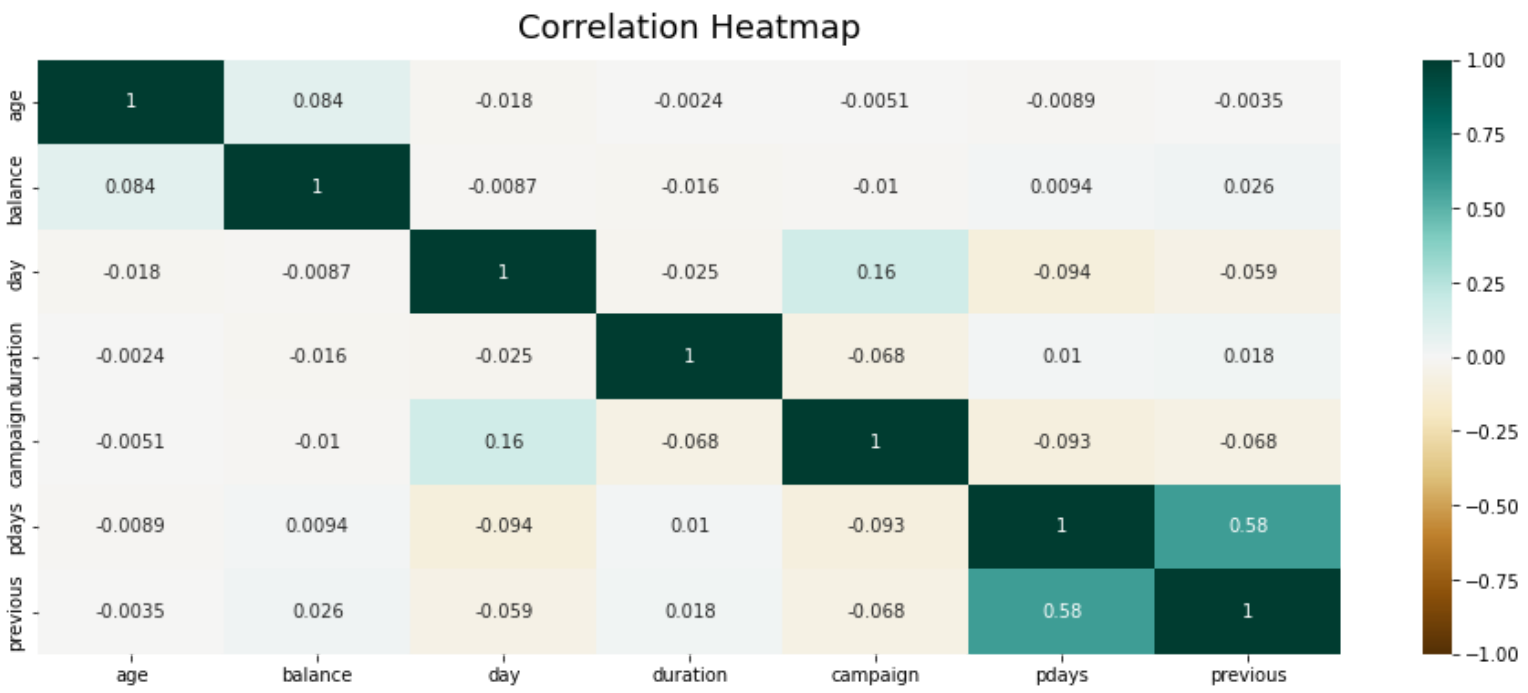
Los datos están relacionados con campañas de marketing directo de una entidad bancaria portuguesa. Las campañas de marketing se basaron en llamadas telefónicas. A menudo, se requería más de un contacto con el mismo cliente, para poder acceder a si el producto (depósito a plazo bancario) estaría (o no) suscrito.

variables

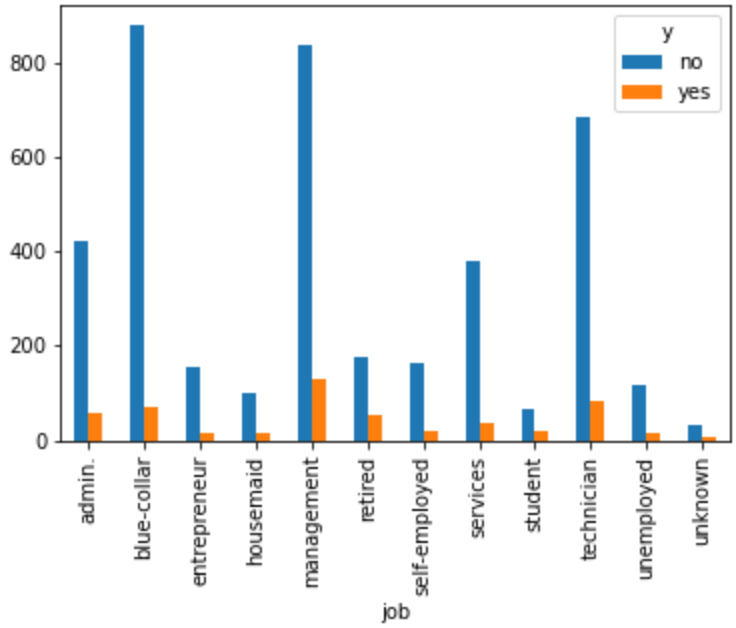
La data corresponde a una serie de variables categoricas y numericas de tal forma que la variable de respuesta es de tipo binaria lo cual nos permite usar el machine learning para modelos de clasificacion

Analisis exploratorio

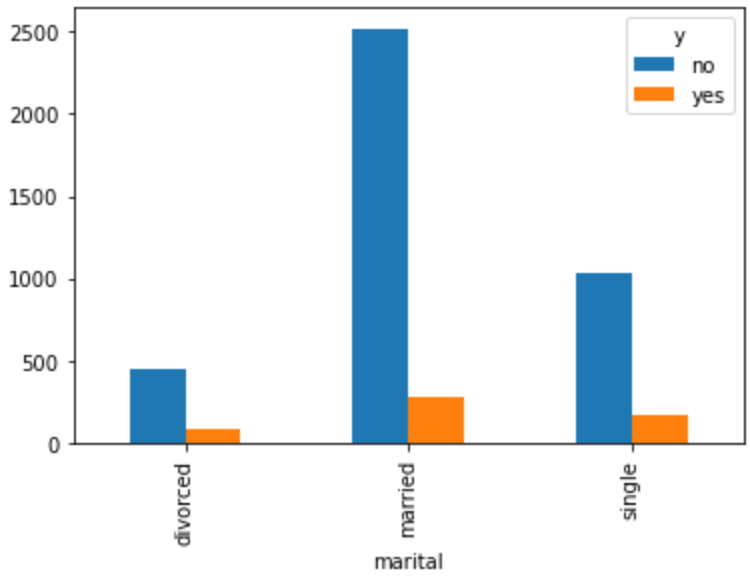

age
balance
day
duration
campaign
pdays
previous
y
no
yes



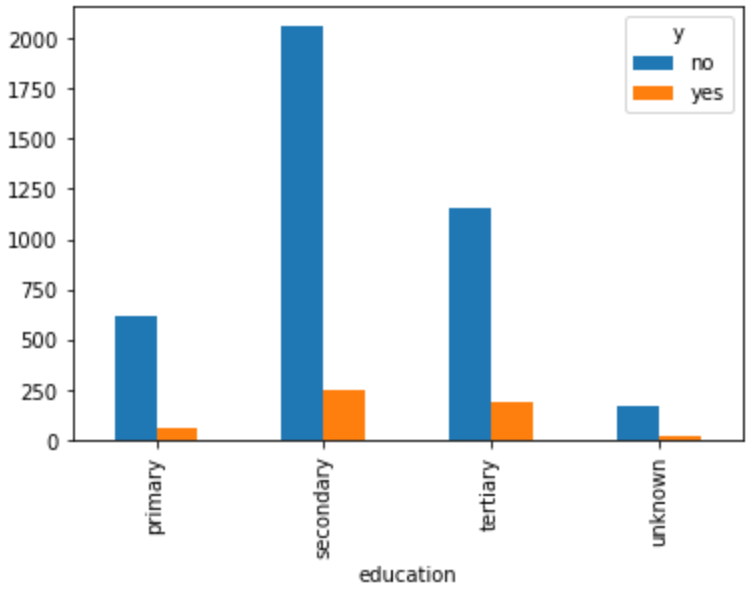

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



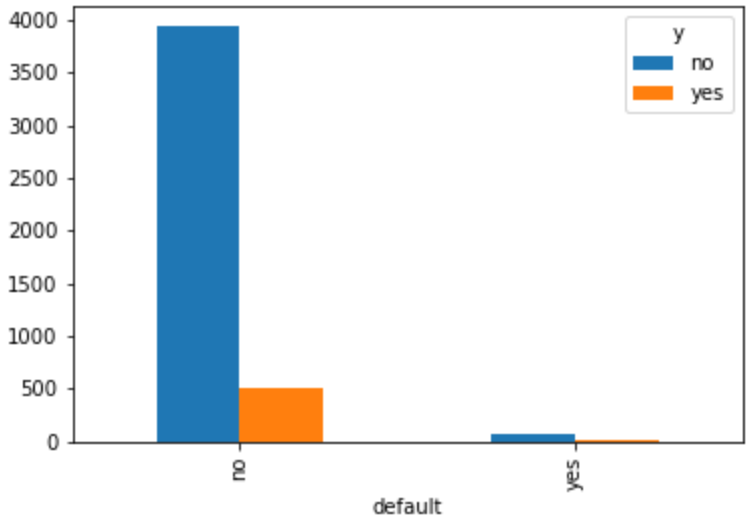

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



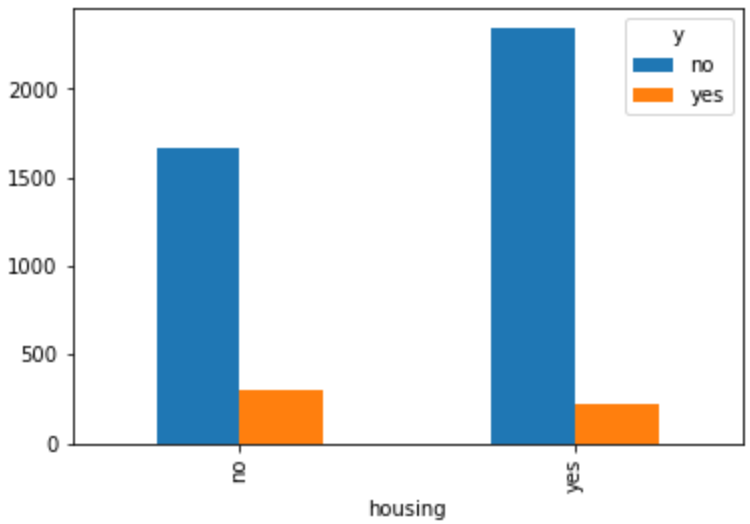
```
AxesSubplot(0.125,0.125;0.775x0.755)
```



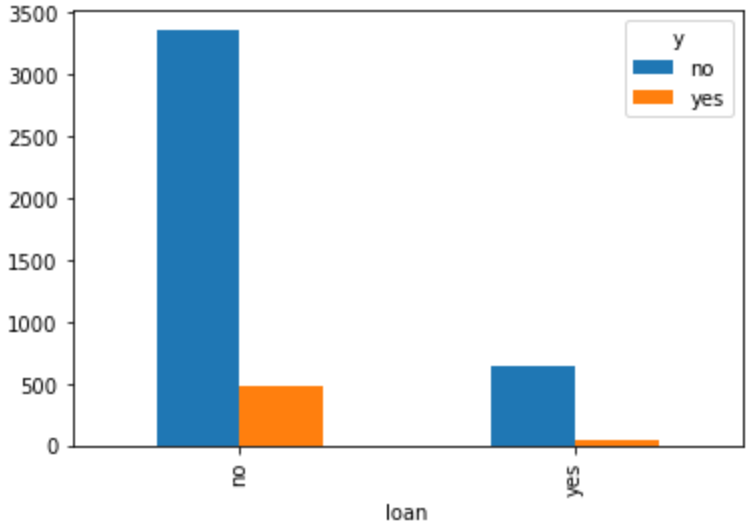

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



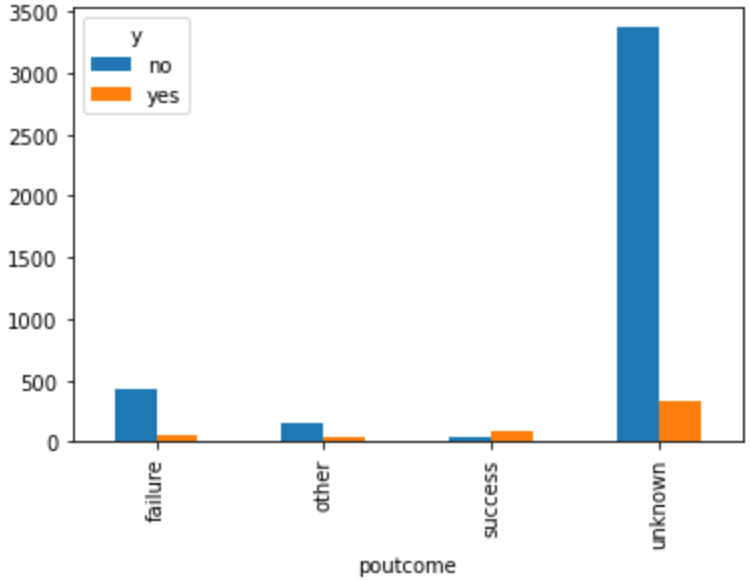

```
AxesSubplot(0.125,0.125;0.775x0.755)
```




```
AxesSubplot(0.125,0.125;0.775x0.755)
```




```
AxesSubplot(0.125,0.125;0.775x0.755)
```



SVM

modelo base

SVC

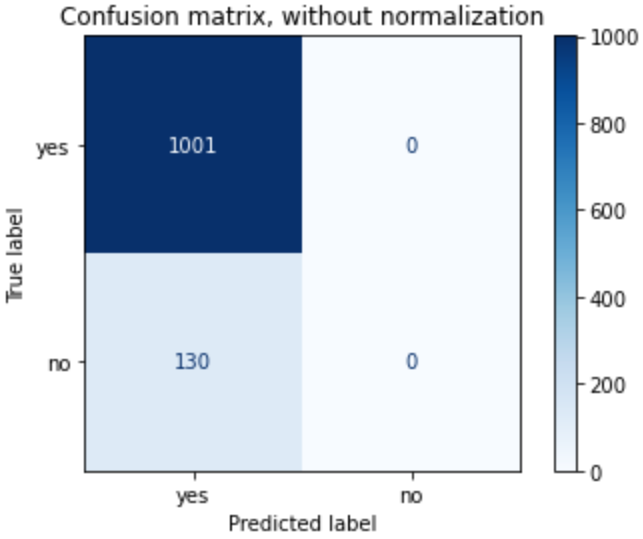
```
SVC(gamma='auto')
```

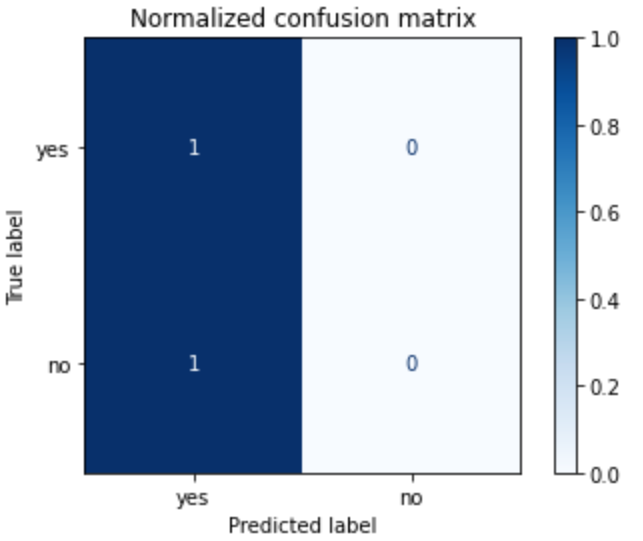
```
El accuracy de test es: 88.50574712643679%
```

matriz de confusion

```
Confusion matrix, without normalization
[[1001    0]
 [ 130    0]]
```

```
Normalized confusion matrix
[[1. 0.]
 [1. 0.]]
```



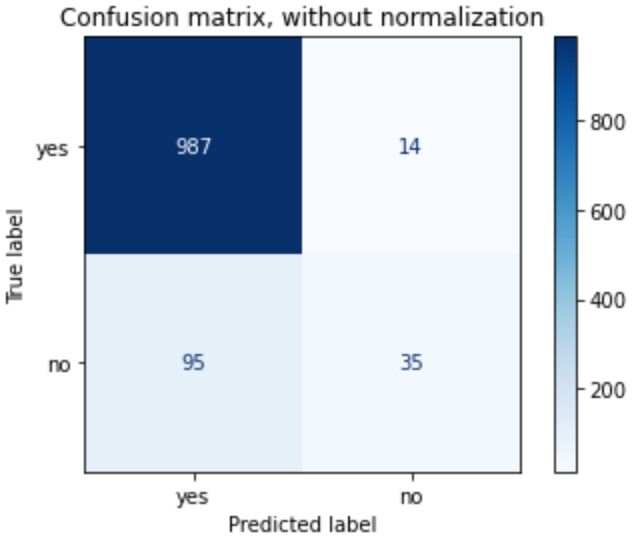


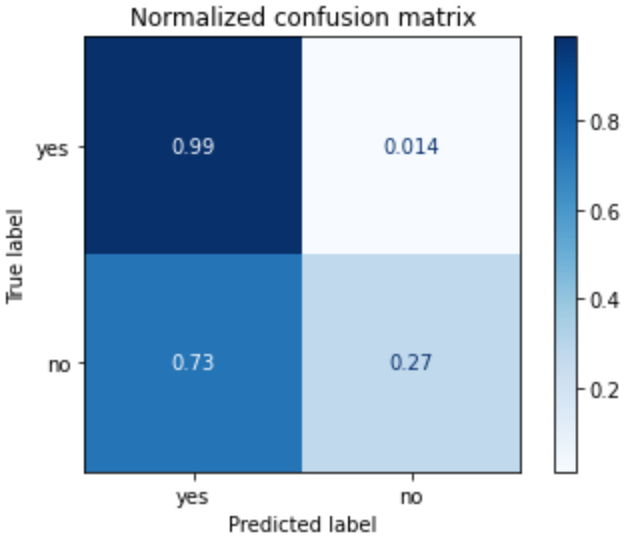
modelo optimizado

```
El accuracy de test es: 90.36251105216623%
```

matriz de confusion

```
Confusion matrix, without normalization
[[987  14]
 [ 95  35]]
Normalized confusion matrix
[[0.99 0.01]
 [0.73 0.27]]
```





Random forest

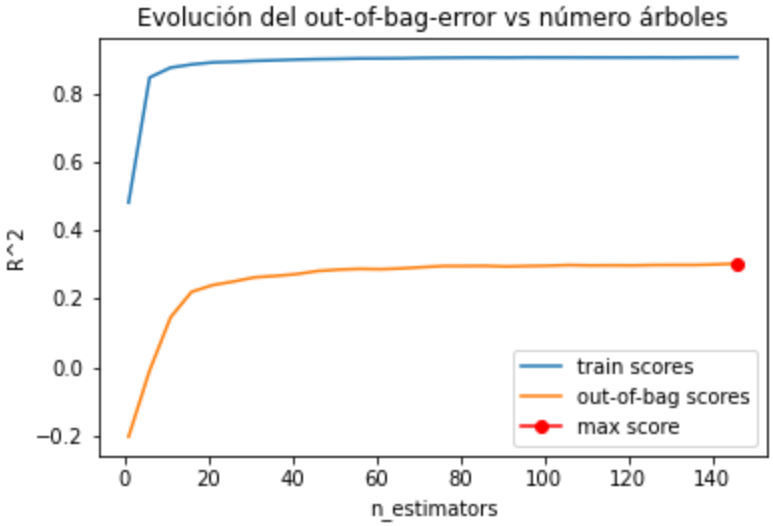
RandomForestRegressor

```
RandomForestRegressor(criterion='mse', max_features='auto', n_estimators=10,
                      n_jobs=-1, random_state=123)
```

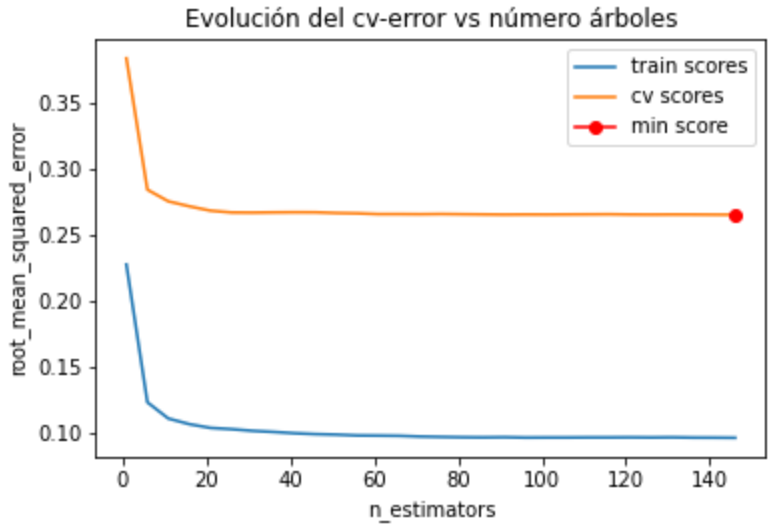
```
El error (rmse) de test es: 0.28743483527642705
```

optimizacion del modelo

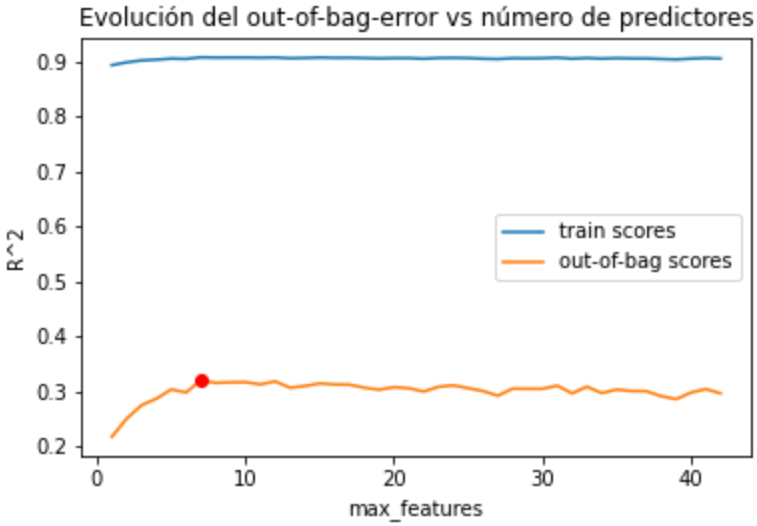
```
Valor óptimo de n_estimators: 146
```



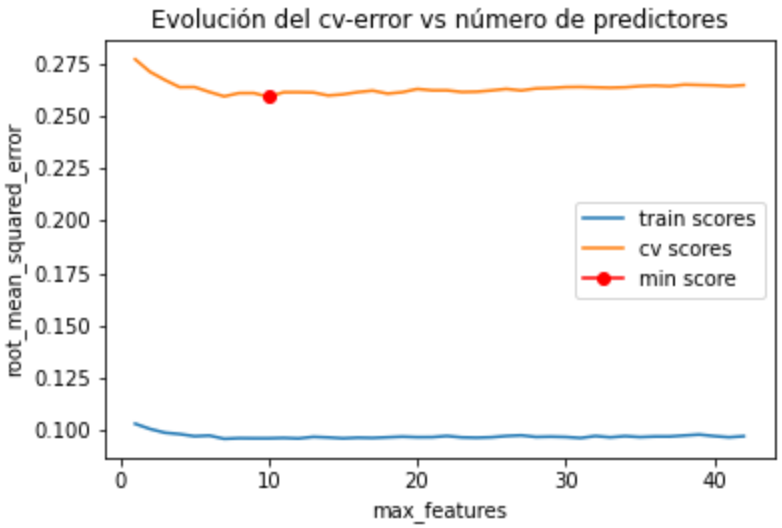

```
Valor óptimo de n_estimators: 146
```




```
Valor óptimo de max_features: 7
```




```
Valor óptimo de max_features: 10
```




```
{'max_depth': None, 'max_features': 11, 'n_estimators': 146}

El nuevo error (rmse) de test es: 0.27922656563195436
versus el modelo estandar aplicado anteriormente: 0.28743483527642705

Se ah conseguido reducir el error en 0.008208269644472688
```

Redes neuronales

modelo base

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_5 (Dense)             (None, 50)                2150

 dense_6 (Dense)             (None, 40)                2040

 dense_7 (Dense)             (None, 40)                1640

 dense_8 (Dense)             (None, 40)                1640

 dense_9 (Dense)             (None, 1)                 41

=================================================================
Total params: 7,511
Trainable params: 7,511
Non-trainable params: 0
_________________________________________________________________
```

comportamiento del modelo

```
36/36 [==============================] - 0s 2ms/step
106/106 [==============================] - 0s 2ms/step - loss: 0.2014 - accuracy: 0.9136
[0.20141176879405975, 0.9135693311691284]
```

conclusion

los modelos ML no ibtuvieron los mejores resultados en esta clasificacion por lo cual seria interesentante comparar los resultados con modelos clasicos como un modelo lineal generalizado ej: modelo logit.